Chapter 29:   Version Files

Version files are **UPS** database files that contain information specific to the local installation and declaration of the declared product instances. The contents of version files are described in this chapter.

29.1 About Version Files

The information in a version file includes (but is not limited to):

- when the instance was declared
- who declared the instance
- the product root directory of the instance
- the location of the `ups` directory
- the location of the table file for the instance

One version file must exist for each version of a product that is declared to the **UPS** database. For a particular version of a product, there is often a separate product instance installed for each flavor; and sometimes more than one per flavor if qualifiers are used. A new version file is created automatically by **UPS** when the first instance of a new version of a product is declared to the **UPS** database via the **`ups declare`** command. When a subsequent instance of the same version is declared, **UPS** automatically modifies the existing version file to include information for it. Multiple product instances are therefore often represented in a single version file.

The naming convention for version files is the version number followed by `.version`, e.g., `v19_34.version`. The version file must reside in the appropriate product-specific directory under the **UPS** database directory, `$PRODUCTS/<product>/<version>.version` (e.g., `$PRODUCTS/emacs/v19_34.version`).

The information in version files is stored in keyword definitions as described in 28.2 *Keywords: Information Storage Format*. The keywords get set according to the options specified on the **`ups declare`** command line.

29.2 Keywords used in Version Files

This is a subset of the list given in section 28.4 *List of Supported Keywords*.

Keyword and Default Value (if any)	Description and Notes (if any)
ARCHIVE_FILE	archive file name/location; used by **UPD**
AUTHORIZED_NODES Default: All nodes (*); taken from **UPS** database configuration file	authorized nodes
COMPILE_DIR	directory in which the compile file resides
COMPILE_FILE	the name of the file containing compiled functions (see Chapter 38: *Use of Compile Scripts in Table Files*)
DECLARED Default: current date and time	the date/time that the instance was declared to **UPS** or declared with a chain Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)
DECLARER Default: current user	userid of user that performed the declaration Note: often has multiple values, one for each declaration (e.g., for subsequent chain declarations)
DESCRIPTION	product description
FILE	type of file (possible values: DBCONFIG, UPDCONFIG, CHAIN, VERSION, TABLE)
FLAVOR	product instance flavor Note: To easily accommodate flavor-neutral **`setup`** functions in a table file, FLAVOR can take the value ANY, but *only* in a table file.
MODIFIED Default: Current date/time	last time the associated instance was changed Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)
MODIFIER Default: Current user	userid of user that modified the instance Note: often has multiple values, one for each declaration/modification (e.g., for subsequent chain declarations)
ORIGIN	master source file; see option **`-D`** in Chapter 25: *Generic Command Option Descriptions*
PRODUCT	product name
PROD_DIR	product root directory (usually defined relative to PROD_DIR_PREFIX, below)

Keyword and Default Value (if any)	Description and Notes (if any)
QUALIFIERS	additional instance specification information often used to indicate compilation options used by developer Notes: appears immediately after a FLAVOR in these files, and is coupled with it to complete the instance identification (see 27.2.3 *Qualifiers: Use in Instance Matching*)
STATISTICS	flag to record statistics for specified products See section 12.8.3 *Collecting Statistics on Product Usage* for usage information.
TABLE_DIR Default: search path (see section 29.4 *Determination of ups Directory and Table File Locations*)	location of table file
TABLE_FILE	name of table file (relative to TABLE_DIR)
UPS_DB_VERSION	**UPS** database version
UPS_DIR Default: `${UPS_PROD_DIR}/ups` if directory exists there	location of `ups` directory (if not absolute path, then taken relative to PROD_DIR, if specified)
VERSION	product version

29.3 Version File Examples

29.3.1 Sample Version File for exmh v1_6_6

Let's declare a new version of **exmh** via the command:

```
% ups declare -r /afs/fnal.gov/products/UNIX/exmh/v1_6_6 \
  -m exmh.table exmh v1_6_6
```

This example assumes the `ups` directory resides in its default location (directly under product root directory), the table file resides in a default location (see section 29.4 *Determination of ups Directory and Table File Locations*) and we are using $PRODUCTS to determine the database (**`-U <upsDir>`**, **`-M <tableFileDir>`** and **`-z <databaseList>`** are unspecified).

Given a machine of flavor SunOS+5, this creates the following version file, named `v1_6_6.version`:

```
FILE = version
PRODUCT = exmh
VERSION = v1_6_6

#*************************************************
#
FLAVOR = SunOS+5
QUALIFIERS = ""
  DECLARED = 1998-03-30 21.06.59 GMT
  DECLARER = stolz
  MODIFIED = 1998-03-30 21.06.59 GMT
  MODIFIER = stolz
  PROD_DIR = /afs/fnal.gov/products/UNIX/exmh/v1_6_6
  UPS_DIR = ups
  TABLE_FILE = exmh.table
```

29.3.2 Sample version file for foo v2_0

Version files can contain information for multiple instances of a single version of a product. Here is an example for a fictional product **foo** v2_0. The file below would have been created and modified by the series of commands:

```
% ups declare foo v2_0 -m v2_0.table -f IRIX -q superoptimize \
  -r /usr/prod/IRIX/foo/v2_0s
```

```
% ups declare foo v2_0 -m v2_0.table -f OSF1 \
  -r /usr/prod/OSF1/foo/v2_0
```

```
FILE = version
PRODUCT = foo
VERSION = v2_0

#*************************************************
#
FLAVOR = IRIX
QUALIFIERS = "superoptimize"
  DECLARER = aheavey
  DECLARED = 1998-04-15 16.37.58 GMT
  MODIFIER = aheavey
  MODIFIED = 1998-04-15 16.37.58 GMT
  PROD_DIR = /usr/prod/IRIX/foo/v2_0s
  UPS_DIR = ups
  TABLE_FILE = v2_0.table

 #----------------------------------------
```

```
#
FLAVOR = OSF1
QUALIFIERS = ""
 DECLARER = aheavey
 DECLARED = 1998-04-15 16.39.58 GMT
 MODIFIER = aheavey
 MODIFIED = 1998-04-15 16.39.58 GMT
  PROD_DIR = /usr/prod/OSF1/foo/v2_0
  UPS_DIR  = ups
  TABLE_FILE = v2_0.table
```

29.4 Determination of ups Directory and Table File Locations

In a version file, the TABLE_DIR and UPS_DIR keywords can each be specified as an absolute or a relative path. When either is specified as a *relative* path, it is taken as relative to PRODUCT_DIR_PREFIX/PRODUCT_DIR[1].

The table file name and directory can be specified in several ways, depending on how their corresponding keywords have been defined. **UPS** uses the following algorithm to determine the table file location:

If TABLE_FILE is specified as an absolute path, then:

- The location is TABLE_FILE.

If TABLE_FILE is specified as a relative path, or simply as the filename, then:

- If TABLE_DIR is specified, the location is TABLE_DIR/TABLE_FILE.
- If TABLE_DIR is not specified, and UPS_DIR is specified, then two locations are searched: first the product subdirectory in the database (e.g., `$PRODUCTS/<product>`), and second UPS_DIR.
- If neither TABLE_DIR nor UPS_DIR is specified at all, **UPS** will search for TABLE_FILE under the product subdirectory in the database only.

[1] Be aware that PROD_DIR_PREFIX may not be defined; if not, PROD_DIR should be an absolute path.